NORTH-HOLLAND

Author Index

A
Abdel-Hamid, T., 38:263
Achee, B.L., 39:179
Agarwal, R., 37:5
Allen, E.B., 36:103
Arai, Y., 38:13
Arnesen, K., 38:85
Azadegan, S., 39:265

B
Balloun, J., 37:181
Basili, V.R., 36:17, 38:3, 38:211
Behder, B., 36:33
Bernardeschi, C., 39:249
Bertolino, A., 38:47
Boloix, G., 36:297
Bolonga, S., 38:1
Borgersen, P.C., 38:85
Brandt, S.J., 38:255
Briand, L.C., 36:313

C
Carver, D.L., 39:179
Cerpa, N., 36:3
Chang, C.-C., 36:167, 36:267
Chang, R.-C., 39:61
Chu, P.-C., 37:187
Chung, C.-G., 39:145
Coppola, P., 38:95
Cuthill, B., 37:27

D
Davis, J.S., 38:155
De Bosschere, K., 39:71
de Figueiredo, J.C.A., 39:39
Delis, A., 38:211
Desharnais, J.-M., 39:281
Dolado, J.J., 37:161

E
Elboushi, M.I., 38:133
Evanco, W.M., 38:27

F
Fantechi, A., 39:249
Finnie, G.R., 39:281
Frankl, P.G., 38:235
Furuyama, T., 38:13

G
Geigel, J., 36:181
George, K.M., 36:147
Glass, R.L., 36:1, 36:85, 36:101, 36:203, 37:1, 37:99, 37:177, 38:105, 38:195, 39:1, 39:83, 39:91, 39:197
Gnesi, S., 39:249
Gorla, N., 36:191
Goscinski, A., 37:201

H
Hakuta, M., 37:253, 38:145
Hardgrave, B.C., 37:19
Hariri, S., 37:215
Harrold, M.J., 36:273
Hausen, H.-L., 38:71
Hedrick, G.E., 36:147
Heiat, A., 39:7
Heiat, N., 39:7
Heikkilä, M., 37:117
Honavar, V., 37:61
Hsu, J.-M., 36:115
Hu, C., 38:235
Huang, C.-M., 36:115
Huang, D.-T., 36:115
Hung, S.-I., 38:119

I
Iio, K., 38:13

J
Jan, J.-K., 39:93
Jiang, J.J., 37:181
Jirachiefpattana, A., 39:119

K
Kameas, A., 36:233
Kanet, J.J., 38:155
Kawasaki, T., 38:37
Kee, K.K., 37:215
Khoshgoftaar, T.M., 36:103
Kim, H.C., 39:103
Klein, G., 37:181
Koriem, S.M., 36:247

L
Lai, R., 36:69, 36:137, 37:75, 37:235, 39:119, 39:223
Laitinen, K., 37:117
Lakhotia, A., 36:211
Lam, K-w, 38:119
Lam, K-y, 38:119
Lam, K.-Y., 37:41
Lam, W., 38:197
Lanning, D.L., 36:103
Lanubile, F., 38:225
Lee, A.Y., 36:33
Lee, G.-S., 37:141
Lee, J.-G., 39:145
Lee, J.-S., 37:141
Lei, S.T., 39:27
Leung, Y.-W., 37:129
Li, K.-C., 39:27
Li, X., 39:223

J. SYSTEMS SOFTWARE 1997; 39:293–294
© 1997 Elsevier Science Inc. All rights reserved.
655 Avenue of the Americas, New York, NY 10010

0164-1212/97/$17.00
PII S0164-1212(97)00154-4

Lin, C.Y., 38:263
Lin, Y.B., 36:181

M

Maeda, N., 38:37
Mashiko, Y., 36:17
Mata-Toledo, R.A., 39:3
Matzen, R.W., 36:147
McQueen, R., 38:165
Mehta, D.P., 39:201
Mikler, A.R., 37:61
Mirandola, R., 38:47

N

Nair, V.S.S., 39:103
Neilsen, M.L., 37:91
Ni, D.C.-L., 39:15

O

Obara, E., 38:37
Offutt, A.J., 36:273
Ohminami, M., 37:253, 38:145
Ookawa, Y., 38:37
Ozkul, T., 37:227

P

Palma, J., 39:165
Panaroni, P., 38:1
Patnaik, L.M., 36:247
Peciola, E., 38:47
Pennington, N., 36:33
Perkusich, A., 39:39
Pintelas, P., 36:233
Pong, J.-C., 36:115
Prasad, J., 37:5

R

Ramakrishnan, R., 36:191
Ravoor, S.B., 38:107
Rowe, N.C., 37:117

S

Sahni, S., 39:201
Salkield, T., 37:41
Schneberger, S.L., 37:101
Sherif, J.S., 38:133, 38:263
Stålhane, T., 38:85
Subramanian, G.H., 38:255

T

Tai, C.-C., 39:93
Takahashi, R., 38:61
Taramaa, J., 37:117
Tewary, K., 36:273
Thomas, W.M., 36:313, 38:211
Tian, J., 39:165
Tone, F., 37:253
Tripathi, A., 39:265
Troster, J., 39:165
Troy, D., 38:165
Tsai, H.-M., 36:267

U

Ulusoy, Ö, 37:49

V

Verner, J.M., 36:3
Visaggio, G., 38:225

W

Wang, C.-H., 36:167
Wang, H.-H., 39:61
Wang, Y.-M., 39:61
Weiss, S.N., 38:235
Welzel, D., 38:71
Willis, M.A., 39:3
Wittig, G.E., 39:281
Wong, J.S.K., 37:61, 38:107

Y

Yaverbaum, G.J., 38:255

Z

Zelkowitz, M., 37:27
Zhang, K., 39:27
Zhou, W., 37:201

NORTH-HOLLAND

Subject Index

A

Abbreviation, source program elements and, 37:117
Acquisition loop time, evaluation/improvement, 37:227
Actors and scripts, object-oriented simulation of enterprise systems and, 37:187
Albrecht Function Points, relationships between Mark II Function Points and, 37:161
Algorithms
  chaos game, random sequences visualization and, 39:3
  efficient communication, ring multicomputers and, 37:215
  loop scheduling, non-uniform memory access (NUMA) multiprocessors and, 39:61
  parallel join, SIMD architectures and, 39:265
  pattern discovery in discrete objects and, 39:201
  quorum-based mutual exclusion, k-coteries and, 37:91
All-uses testing, mutation vs., 38:235
Alternating Bit protocol, Estelle to Erlang and, 37:235
American companies, management style, 38:105
American programmers, software reliability and, 38:105
Artificial neural networks, effort estimation and, 39:281
Authoring systems, subsystems, 36:233
Automatic code generation, large-scale, 37:1
Automatic communication protocol implementation, compilation method for, 37:235
Automation, pattern discovery and, 39:201
Automation measurement, Information Technology Engineering and Measurement (ITEM) Model and, 37:27
Availability, distributed systems and, 38:107

B

Blocking message passing assumptions, communication function design and implementation and, 37:215
Branch coverage, test estimate for, 38:47
Branch testing automation, case study, 38:47

C

C training, COBOL programmers and, 37:5
Case-based reasoning, effort estimation and, 39:281
CASE tools, Domain specific, development and evaluation, 38:165
Certification, software evaluation for, 38:71
Chaos game algorithms, random sequences visualization and, 39:3
Chinese data files, data compression strategy and, 36:167
Classification mode, software quality, 38:61
Classification techniques, software subsystem, 36:211
Client server architechture, new technology and, 37:99
Clustered affinity scheduling, non-uniform memory access (NUMA) multiprocessors and, 39:61
COBOL, demise predictions, 39:197
COBOL programmers, C training and, 37:5
Code generation, automatic, 37:1
Communication protocol, automatic compilation method, 37:235
  verification, 39:235
Completeness, software design, 36:33
Complex protocol, verification, 36:69
Complexity
  Information Technology Engineering and Measurement (ITEM) Model and, 37:27
  Poisson analyses of defects, 38:27
  software maintenance and, 37:101
  software quality classification model and, 38:61
Component quality, predictive quality models and, 38:225
Component reuse, errors analysis, 38:211
Component simplicity, software maintenance and, 37:101
Compression strategy, Chinese-English characters and, 36:167
Computation time, zero-one optimal path set selection method and, 39:145
Computer Operations Performance System (COPS), flexible graphical user interface and, 36:181
Computer science
  history, 36:203
  Ph.D. statistics, 39:91
  scholars/institutions, survey findings, 39:1, 39:83
Concurrency control protocol, real-time databases and optimistic, 38:119
Concurrent systems, verification with JACK, 39:249
Conference, Third International Conference on Achieving Quality in Software (AQuIS'96), 38:1
Congestion control, Quo Vadis and, 37:61
Control, programmer discipline and, 36:101
Corporate Information Management (CIM) interface architecture, Information Technology Engineering and Measurement (ITEM) Model and, 37:27
Costs
  controlling software, 36:191
  portability and, 38:145
Cryptographic Timed Petri Net. *See* CTPN
CSMA/CD protocol, performance results, 37:75
CTPN, cryptographic protocols and, 37:141
Customer's view, software quality and, 38:85

D

Data acquisition, acquisition loop time and, 37:227
Data broadcast, design and implementation, 37:215
Data collection, design and implementation, 37:215
Data compression strategy, Chinese-English characters and, 36:167
Data distribution, design and implementation, 37:215
Databases
  availability, directory services and, 37:41
  metric, software complexity metric and, 36:103
  real-time
    network access protocols and, 37:49
    optimistic concurrency control protocol and, 38:119
Date crisis problem, solutions, 37:177
Decentralization, software maintenance and, 37:99, 37:101
Deductive Machine, ISO ACSE protocol verification, 39:223
Defect prediction, Poisson regression techniques and, 38:27

J. SYSTEMS SOFTWARE 1997; 39:295–299
© 1997 Elsevier Science Inc. All rights reserved.
655 Avenue of the Americas, New York, NY 10010

0164-1212/97/$17.00
PII S0164-1212(97)00155-6

Delay
propagation, communication function design and implementation and, 37:215
Quo Vadis and message, 37:61
real-time database systems and network, 37:49
Design analysis, Quality Function Deployment and, 38:133
Development, object-oriented, 37:19
Development environment, GENITOR, 36:233
Directory service, implementation, 37:41
Disabbreviation, source program elements and, 37:117
Discipline, programmer, 36:101
Distributed object based computing, fault tolerance and, 39:103
Distributed systems
fault-tolerant servers for the RHODOS system, 37:201
k floating software licenses and, 37:91
multithreaded transaction processing and, 38:107
network access protocols and, 37:49
Document definition, Standard Generalized Markup Language (SGML) and, 36:147
Document design, Standard Generalized Markup Language (SGML) and, 36:147
Domain analysis techniques, requirements reuse and, 38:197
Domain/computation fault classification scheme, fault modeling/seeding and, 36:273
Domain specific CASE tools, development and evaluation, 38:165
Dummy variables, regression analysis for software modeling and, 36:313
Dynamic resource-allocation, software-module testing and, 37:129

E

Editorial policy, "nonchange", 36:1
Effort estimation
function points and, 39:281
model for small-scale business applications, 39:7
Elections, secure electronic voting protocol with IC cards 39:93
Electronic voting protocol, IC cards and, 39:93
Element declarations, Standard Generalized Markup Language (SGML) and, 36:147
Engineering scholars/institutions
Ph.D. statistics, 39:91
survey findings, 39:1, 39:83
English data files, data compression strategy and, 36:167
Enterprise systems, object-oriented simulation and, 37:187
Enumeration tools, WINDOWS and UNIX environments and, 39:15
Environmental factors, software size estimation model and, 37:253
Erlang, semantic gap between Estelle and, 37:235
Errors analysis, reuse-oriented development environment and, 38:211
Estelle
Estelle Development Toolset, 36:137
EVEN, 39:119
protocol design system and, 36:115
translation into Erlang codes, 37:235
Estelle Verification Environment using NPNs (EVEN). *See* EVEN
Estimation model, software size, 37:253
European companies, management style, 38:105
Evaluation framework, software systems, 36:297
EVEN, specification verification, 39:119
Expert systems, effectiveness evaluation, 38:255

F

Fault generation, stress and, 38:13
Fault modeling, program dependence graph and, 36:273
Fault-seeding tools, program dependence graph and, 36:273
Fault tolerance
distributed object based computing and, 39:103
distributed systems and, 38:107
Fault-tolerant servers, RHODOS system and, 37:201
Faults
dynamic resource allocation for software-module testing and, 37:129
prediction techniques for component quality and, 38:225
Flexible graphical user interface, network simulation and, 36:181
Formal Description Techniques
Estelle-based incremental protocol design system, 36:115
Estelle translation to Erlang, 37:235
protocol development and, 36:137
protocol specification and, 36:69
Formal language model, Standard Generalized Markup Language (SGML) and, 36:147
Formal methods, protocol development and, 36:69
Formal specification and analysis, cryptographic protocols and, 37:141
Forms-based tools, requirements reuse and, 38:197
Function Points
effort estimation and, 39:7, 39:281
relationships between Albrecht and Mark II, 37:161
Functional design methodologies, stress effects on fault generation and, 38:13
Functionality, tradeoffs with schedule, quality, and, 38:105

G

G-nets, logical and timing analysis of complex systems, 39:39
GENITOR, functional architecture and interaction model of, 36:233
Goal/Question/Metric (GQM) paradigm, software process improvement and, 36:17
Graphical approach, production scheduling with interactive, 38:155
Graphical techniques, random sequences and, 39:3
Graphical user interface, guidelines and implementation, 36:181
GreatSPN, CSMA/CD protocol performance results and, 37:75

H

Hardware fault tolerance, distributed object based computing and, 39:103
Hardware selection, acquisition loop time and, 37:227
History, computer science, 36:203

I

IC cards, electronic voting protocol and, 39:93
Incremental protocol design system, verification and, 36:115
Information system evaluation, system typology and, 37:181
Information system life cycle planning, complexity model and, 37:101
Information Technology Engineering and Measurement (ITEM) Model, application in software engineering environments, 37:27
Information technology model, application in software engineering environments, 37:27
Institutions, survey findings, 39:1, 39:83
Intelligent Tutoring Systems, generator, 36:233
ISO ACSE protocol, verification, 39:223

J

JACK environment, industrial application, 39:249
Japanese companies, management style, 38:105
Job insecurity, C training for COBOL programmers and, 37:5
Join operation, SIMD architectures and parallel, 39:265

Journal of Systems and Software, editorial policy, 36:1
Just Another Concurrency Kit (JACK). *See* JACK

K

K-coteries, properties of nondominated, 37:91

L

LAN system, CSMA/CD protocol and, 37:75
Left cycle development process, expert systems effectiveness and, 38:255
Legacy FORTRAN code, object-oriented design creation and, 39:179
Legal users, secret sharing scheme and, 36:267
Lines of code (LOC)-based model, effort estimation and, 39:7
Lines of code (LOC) measure, metrics for replacement of common, 37:161
Logical analysis, complex systems and, 39:39
Loop scheduling algorithms, non-uniform memory access (NUMA) multiprocessors and, 39:61

M

Management
  European and Japanese vs. American companies, 38:105
  programmer discipline and, 36:101
Mark II Function Points, relationships between Albrecht and, 37:161
McCabe's complexity measure, software quality classification model and, 38:61
Mental stress conditions, fault generation and, 38:13
Message delay, Quo Vadis and, 37:61
Message-passing parallel programs, performance tuning tools and, 39:27
Metric database, software complexity metric and, 36:103
Metrics
  Information Technology Engineering and Measurement (ITEM) Model and, 37:27
  performance, Quo Vadis and, 37:61
  software testing and, 38:37
Modelling
  CSMA/CD protocol and performance, 37:75
  pattern discovery in discrete objects and, 39:201
  process improvement through, 38:95
  quality prediction techniques, 38:225
Move-To-Front method, Chinese-English characters and, 36:167
Multicomputers, bidirectional ring, efficient communication algorithms and, 37:215
Multithreaded transaction processing, distributed systems and, 38:107
Mutation testing, all-uses vs., 38:235

N

Name replacements, disabbreviation of source program elements and, 37:117
Network access protocols, real-time database systems and, 37:49
Network delay, real-time database systems and, 37:49
Network directory service, implementation, 37:41
New technologies, productivity/quality and, 36:85
Non-uniform memory access (NUMA) multiprocessors, clustered affinity scheduling and, 39:61

O

Object-oriented concepts, introduction, 37:187
Object-oriented designs, legacy FORTRAN code and, 39:179
Object-oriented simulation, enterprise systems and, 37:187
Object-oriented software design, Quality Function Deployment and, 38:133
Object-oriented technology, development and, 37:19
Optimal path set selection method, zero-one, enhanced method, 39:145
Organizational tenure, C training for COBOL programmers and, 37:5

P

Parallel/distributed systems, petri net model for performance evaluation, 36:247
Parallel join algorithms, SIMD architectures and, 39:265
Parallel logic programming, process-based, 39:71
Parallel programs, performance tuning tools and, 39:27
Parallelism, distributed systems and, 38:107
Parsing methods, Standard Generalized Markup Language (SGML) and, 36:147
Pattern discovery, automation, 39:201
Performance, k-coteries and, 37:91
Performance metrics, Quo Vadis and, 37:61
Performance modelling, CSMA/CD protocol, 37:75
Performance tuning tools, parallel programs and, 39:27
Petri net based approach, logical and timing analysis of complex systems, 39:39
Petri net based models, cryptographic protocols and, 37:141
Petri net theory, performance evaluation of parallel/distributed systems and, 36:247
Ph.D. statistics, computer science/engineering, 39:91
Pilot studies, guidelines, 36:85
Point-to-point routing, communication function design and implementation and, 37:215
Poisson regression techniques, defect prediction and, 38:27
Predicate/transition nets, performance evaluation of parallel/distributed systems and, 36:247
Process-based parallel logic programming, basic issues survey, 39:71
Process improvement, process modelling and, 38:95
Process modelling, process improvement through, 38:95
Process Weaver, process enactment and, 38:95
Processing complexity, software size estimation model and, 37:253
Production scheduling, interactive graphical approach, 38:155
Productivity, new technologies and, 36:85
Products and processes quality, evaluation, 38:71
Program dependence graph, fault modeling/seeding and, 36:273
Programmers
  software reliability and American, 38:105
  software structure attributes effects, 36:191
Project Support Environment (PSE) model, Information Technology Engineering and Measurement (ITEM) Model and, 37:27
Propagation delay, communication function design and implementation and, 37:215
Protocol development, Formal Description Techniques and, 36:137
Protocol specification, Formal Description Techniques and, 36:69
Prototyping, software development and, 36:3

Q

Quality
  new technologies and, 36:85
  prediction techniques for component, 38:225
  tool support and, 39:165
  tradeoffs with functionality, schedule, and, 38:105

Quality assessment, software systems, 36:297
Quality Function Deployment, object-oriented software design and, 38:133
Quality improvement technology, reading technologies and, 38:3
Questionnaire, customer's quality view, 38:85
Queuing model, actors and scripts and, 37:187
Quo Vadis, intelligent routing in large networks and, 37:61
Quorum-based mutual exclusion algorithms, k-coteries and, 37:91

R

Random sequences visualization, chaos game algorithms and, 39:3
Reading technologies, evolving and packaging, 38:3
Real-time database systems
  network access protocols and, 37:49
  network delay and, 37:49
Real-time databases, optimistic concurrency control protocol and, 38:119
Regression analysis
  effort estimation and, 39:281
  software modeling and, 36:313
Reliability
  distributed systems and, 38:107
  fault generation caused by stress and, 38:13
  k-coteries and, 37:91
Remote Procedure Calls, distributed systems and, 38:107
Requirements acquisition, Quality Function Deployment and, 38:133
Requirements reuse, achieving, 38:197
Resource allocation, software-module testing and dynamic, 37:129
Resource sharing, distributed systems and, 38:107
Reuse
  component, 38:211
  requirements, 38:197
RHODOS system, fault-tolerant servers and, 37:201
Ring multicomputers, efficient communication algorithms and, 37:215

S

Scalability, distributed systems and, 38:107
Schedule
  interactive graphical approach to production, 38:155
  tradeoffs with functionality, quality, and, 38:105
Schedules, clustered affinity, non-uniform memory access (NUMA) multiprocessors and, 39:61
Scholars/institutions, survey findings, 39:1, 39:83
Scholarship, survey findings, 39:1, 39:83
Scoring system, software design, 36:33
Secret sharing scheme, cryptanalysis of generalized, 36:267
Security, electronic voting protocol with IC cards, 39:93
SGML. *See* Standard Generalized Markup Language (SGML)
Shadow holder, secret sharing scheme and, 36:267
Sharing policy, secret sharing scheme and, 36:267
SIMD architectures, parallel join algorithms and, 39:265
Simulation, resource allocation, 37:129
Simulation tools, flexible graphical user interface and, 36:181
Size estimation model, evaluation of software, 37:253
Small-scale business applications, estimating efforts model, 39:7
Software complexity, Poisson analyses of defects, 38:27
Software costs
  controlling, 36:191
  portability and, 38:145
Software design
  authoring systems and, 36:233
  object-oriented, Quality Function Deployment and, 38:133
  scoring system, 36:33
Software development
  distributed systems and software maintenance, 37:101
  effort estimation, 39:7, 39:281
  fault generation caused by stress, 38:13
  fault modeling/seeding, 36:273
  programmer discipline and, 36:101
  prototyping vs. waterfall approaches, 36:3
  software structure attributes effects, 36:191
  tool support and, 39:165
Software engineering, scholars/institutions, survey findings, 39:1, 39:83
Software engineering environments, information technology model application, 37:27
Software-Engineering Process Simulation Model (SEPS), description, 38:263
Software fault tolerance, distributed object based computing and, 39:103
Software field, history, 36:203
Software life cycles
  pilot studies and, 36:85
  time-honored concepts and, 38:195
Software maintenance
  decentralization and, 37:99, 37:101
  disabbreviation of source program elements and, 37:117
Software measurement, tool support and, 39:165
Software metric, quality improvements and, 36:103
Software modeling, regression analysis and, 36:313
Software-module testing, dynamic resource-allocation and, 37:129
Software portability, cost estimation, 38:145
Software process improvement, Goal/Question/Metric (GQM) paradigm and, 36:17
Software productivity, programmer discipline and, 36:101
Software project development process, simulation model, 38:263
Software quality
  American programmers and, 38:105
  classification mode, 38:61
  customer's view, 38:85
  fault generation caused by stress and, 38:13
  Poisson analyses of defects, 38:27
  process improvement, 38:95
  reading technologies and, 38:3
  software evaluation for certification, 38:71
  Third International Conference on Achieving Quality in Software (AQuIS'96), 38:1
  tool support and, 39:165
Software selection, acquisition loop time and, 37:227
Software size estimation model, evaluation, 37:253
Software structure attributes, effects on software development productivity, 36:191
Software subsystem classification techniques, unified framework for expressing, 36:211
Software systems, evaluation framework, 36:297
Software testing
  all-uses vs. mutation testing, 38:235
  fault modeling/seeding, 36:273
  prediction/improvement, 38:37
  zero-one optimal path set selection method, 39:145
Source program elements, disabbreviation and, 37:117
Space-on-board project, process improvement and, 38:95
Specification and analysis, cryptographic protocols and, 37:141
Specification verification, EVEN and, 39:119
Standard Generalized Markup Language (SGML), formal language model for parsing, 36:147
Standardized formal specification, protocol development and, 36:137

Stress, fault generation and, 38:13
Structured design methodologies, stress effects on fault generation and, 38:13
Subsystems
authoring process, 36:233
classification techniques, 36:211
Surveys
process-based parallel logic programming, 39:71
scholars/institutions, 39:1, 39:83
Switching software, test prediction/improvement and, 38:37
System complexity, software maintenance and, 37:101
System typology, information system evaluation and, 37:181
Systems effectiveness, expert, left cycle development process and, 38:255

T

Technologies, new
client server architecture, 37:99
productivity/quality and, 36:85
Technology transition, Information Technology Engineering and Measurement (ITEM) Model and, 37:27
Tenure, organizational, C training for COBOL programmers and, 37:5
Test productivity, prediction/improvement, 38:37
Time-honored concepts, software life cycles and, 38:195
Timing analysis, complex systems and, 39:39
Tool support, software measurement, analysis, and quality improvement, 39:165
Traceability tools, WINDOWS and UNIX environments and, 39:15
Traffic management, intelligent, 37:61
The Transaction Processing Draft International Standard, protocol development and, 36:137
Twin-servers model, fault-tolerant servers for RHODOS system and, 37:201

U

UNIX, enumeration and traceability tools, 39:15

V

Verification, ISO ACSE protocol, 39:223
Verification specification, EVEN and, 39:119
Verification techniques
complex protocol and, 36:69
Estelle-based incremental protocol design system, 36:115
Verification tools, JACK environment and, 39:249

W

Waterfall approach, software development and, 36:3
WINDOWS, enumeration and traceability tools, 39:15

Z

Zero-one optimal path set selection method, enhanced method, 39:145